**TITLE IS CENTERED AND WRITTEN IN ALL CAPS: TITLES THAT EXTEND BEYOND ONE LINE ARE DOUBLE SPACED**

A Second Year Project Report

Submitted to the Faculty

of the

Bennett University

By

[Student’s full name separated by comma]

[Student’s enrollment number separated by comma]

A close up of a sign

Description automatically generated

Department of Computer Science Engineering

November 2019

Greater Noida-201310, Uttar Pradesh, India

**TABLE OF CONTENTS**

<<Right click on the heading and click update flied. Your heading will be pulled here. Same goes with all the table of content in the subsequent pages>>

[LIST OF TABLES v](#_Toc32826715)

[LIST OF FIGURES vi](#_Toc32826716)

[1. INTRODUCTION 1](#_Toc32826717)

[1.1. Problem Statement 1](#_Toc32826718)

[2. Background Research 2](#_Toc32826719)

[2.1. Proposed System 2](#_Toc32826720)

[2.2. Goals and Objectives 2](#_Toc32826721)

[3. Project Planning 4](#_Toc32826722)

[3.1. Project Setup 4](#_Toc32826723)

[3.2. Stakeholders 4](#_Toc32826724)

[3.3. Project Resources 5](#_Toc32826725)

[3.4. Assumptions 5](#_Toc32826726)

[4. SYSTEM ANALYSIS AND DESIGN 7](#_Toc32826727)

[4.1. Overall Description 7](#_Toc32826728)

[4.2. Users and Roles 7](#_Toc32826729)

[4.3. User Stories (Requirements) 8](#_Toc32826730)

[4.4. Design diagrams/ UML diagrams/ Flow Charts/ E-R diagrams 14](#_Toc32826731)

[4.4.1. Use Case Diagrams 14](#_Toc32826732)

[4.4.2. Class Diagram 15](#_Toc32826733)

[4.4.3. Activity Diagrams 16](#_Toc32826734)

[4.4.4. Sequence Diagram 17](#_Toc32826735)

[4.4.5. Data Architecture 17](#_Toc32826736)

LIST OF TABLES

Table Page

[Table 1: Goal and Objectives 2](#_Toc32826670)

[Table 2: Sample 2 4](#_Toc32826671)

[Table 3: Sample 3 4](#_Toc32826672)

[Table 4: Sample 4 5](#_Toc32826673)

[Table 5: Sample 4 5](#_Toc32826674)

LIST OF FIGURES

Figure Page

[Figure 1: Sample use-case diagram 14](#_Toc32826675)

[Figure 2: sample 2 16](#_Toc32826676)

[Figure 3: sample 3 16](#_Toc32826677)

[Figure 4: Sample 4 17](#_Toc32826678)

[Figure 5: Sample 5 18](#_Toc32826679)

1. INTRODUCTION

<<Provide a brief overview of the current trends and situation around your project>>

Paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text paragraph text

* Bulleted list items use the “Bullets” style from the styles pane.
  1. Problem Statement

<< Provide a concise statement on the problem that currently exists and is affecting the organization/society/task.

Example: Currently, Microsoft uses Microsoft Dynamics AX as their enterprise resource planning (ERP) solutions for businesses. The latest version of Microsoft Dynamics AX will collect a significant amount of telemetry about the user’s actions while navigating in the application and its thousands of forms.>>

1. Background Research

<<Provide a detailed description of the literature research that you conducted around your project topic (make sure to cite the relevant sources in your text. E.g., research papers/web articles/blogs) and what motivated you, after researching, to work on the project.

Would suggest you to download a tool (Mendeley: <https://www.mendeley.com>) for automated citation and generation of references.>>

* 1. Proposed System

<< Provide a concise statement on the purpose of the project; the problem or opportunity addressed. The explanation should include what you intended to do. Vision – how will the customer’s world improve as a result of this project? When appropriate, tie this into what is currently being considered or has just been completed at the organization

Example: This project aims to extract information about how users use the forms by analyzing the raw telemetry data. By using big data analysis and machine learning techniques we hope to develop predictions into what actions the user will do next. Using these insights, the Microsoft Dynamics AX engineering may be able to optimize the user experience and reduce the number of steps needed to perform desired actions or reach desired forms.>>

* 1. Goals and Objectives

<< State, in quantifiable terms, if possible, the goals and objectives of the project. Goals may be related to product, process, quality, or teamwork. >>

**Example:**

<<to insert caption for tables and figures, click on references on the top menu, under captions section 🡪 click insert caption 🡪 choose label as table>>

Table 1: Goal and Objectives

|  |  |
| --- | --- |
| **#** | **Goal or Objective** |
| 1 | Make the system extensible – future updates like xxx can be done easily |
| 2 | Make the system easy to support – provide good documentation, configuration/build files, administrator’s manual |
| 3 | Make the system very easy to use – users would agree that minimal to no training is needed |
| 4 | Build a prototype that demonstrates the user interface by xx/xx/xx - in order to get early feedback from the customer/users |
| 5 | Have fun working on the project |

1. Project Planning

This section covers the details of the project planning. Selecting the lifecycle of the development, project stakeholders, resources required, assumptions made (if any) are detailed in the sections below.

* 1. Project Setup

<< Define some of the basic project decisions that will be used on this project. >>

**Example:**

Table 2: Sample 2

|  |  |
| --- | --- |
| **#** | **Decision Description** |
| 1 | Windows 8, C#, OpenSphere vs. Azure, Trac/SVN vs. Git, etc. |
| 2 | Standards that must be followed (default Capstone coding standard, etc.) |
| 3 | Special access privileges needed, nondisclosure forms, release to open source, etc. |
| 4 | A virtual server image will be set up at NDSU that matches the customer environment (image provided by customer) |

* 1. Stakeholders

<< Identify all stakeholders for this project (groups or individuals that are affected by or are in some way accountable for the outcome of the project – business managers, end users, developers, testers, support people, instructors, etc.) >>

**Example:**

Table 3: Sample 3

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| Person A | Sponsor |
| Person B | Mentor |
| Person C | Instructor |
| Person D | Team member |
| Person E | Team member |
| Person F | Team member |

* 1. Project Resources

<< Identify the anticipated resources required for this project. This can include staff members who will work on the project, equipment needed for the project, special software that will need to be acquired, or any other resource necessary for the project. >>

**Example:**

Table 4: Sample 4

|  |  |  |
| --- | --- | --- |
| **Resource** | **Resource Description** | **Quantity** |
| Database Server | A database server provided by the sponsoring company. | 1 |
| Capstone Team | Our team of students who will be the primary developers of the project. | 4 |
| Jim Somebody | The mentor who will be able to provide us with technical assistance. | 1 |
| Mac Workstation | An OS X workstation with X Code for developing the OS X version of the software. | 1 |
| Android Phone | An Android phone to be used as test hardware for the mobile version of the software. | 2 |

* 1. Assumptions

<<State any assumptions upon which the project is based. Assumptions may be related to staffing, resources, tools, and schedules/deadlines. >>

**Example:**

Table 5: Sample 4

|  |  |
| --- | --- |
| **#** | **Assumption** |
| A1 | The capstone team and mentors will be able to meet face to face once a week. |
| A2 | Azure ML will be available for the team to work with as a trial for the first month of the project. |
| A3 | Team members will be able to familiarize themselves with the Azure ML, Azure HDInsights, and R environments |
| A4 | Team will have sufficient time to complete a working model to present by mid-semester |
| A5 | Machine Learning model will be completed in time to test on true big data using HDInsights and Hadoop |
| A6 | The development test data provided will be sufficient to create an accurate prediction of user actions |
| A7 | The models developed will be easily extended to other forms within the time frame |

1. SYSTEM ANALYSIS AND DESIGN
   1. Overall Description

<< Provide a more detailed, two to three paragraph description of the project. This description may include more technical details to describe the purpose of the project.

**Example:** This project is an attempt to apply data science and machine learning techniques to telemetry data from Microsoft ERP products in an attempt to anticipate user actions based on previous navigation and controls to create more efficient application navigation. Using Event Tracer for Windows, test teams are able to record millions of data points with information regarding actions that users have taken, forms they have navigated through, and the time data for each of those events. By combing the data and arranging it by session ID and time we can create a sort of roadmap of each user’s actions in time order from the moment they start the application.

First this data must be cleaned and sorted using R statistical software. Then graph visualizations of the data (initially pertaining to the navigation path through the application) will be generated to allow us to view the dominant paths to specific modules and forms in the product, and give us an idea of where to start with machine learning. The statistics and visualization using R will be compiled into an R package to allow for easy documentation, and extensible use with various datasets. The visualization and statistics functions will provide basic insight into the data for any Program Manager, or someone not experienced in data science. >>

* 1. Users and Roles

<< Provide a list and description of the different types of users or roles within the system. This may include different classes of users, such as administrator, instructor, student, etc. This list may also include autonomous agents that interact with the system as well. These may include users (or personas) that are used as part of any user stories produced for the project. >>

**Example:**

|  |  |
| --- | --- |
| **User** | **Description** |
| Developer | A capstone team member or mentor who is tasked with managing the test data, creating initial machine learning models, and ultimately generating a firm process for applying these techniques to future user data. This is used for sub-stories and task needed to fulfill the true end user use cases. |
| Microsoft Program Manager | A manager at Microsoft who is working on developing the ERP application who will be making design decisions based on the data analysis. |
| Dynamics AX User | An end user of the Microsoft ERP product who will be generating the data used and reaping the potential efficiency benefits from the data analysis when designing the application. |

* 1. User Stories (Requirements)

<< This section lists the user stories for the project, when they were added, and information about which of these user stories the team has committed to complete (C), stories that will be targeted but not guaranteed to be completed (T), and those which the team will not commit to (NC).

It is preferred to have user stories written in the form - *As an [actor] I want [action] so that [achievement].* So, for example: *As a Bison Tracker member, I want to set different privacy levels on my photos, so I can control who sees which of my photos.*

The **ID** column provides a high-level ID for each user story. This is useful for generating acceptance criteria IDs which can be easily associated with a particular user story. The **Added** column lists when the user story was added to the project as it is possible that not all requirements will be elicited or available at the project’s onset. The **Description** column gives a description of the user story. The **Status** column indicates whether these user stories are Committed (i.e. C will be completed by the team), Targeted (i.e. T will be completed if the team has time after all committed requirements are completed), or Not Committed (i.e. NC will not be completed by the team). The **Story Points** indicate how much work is associated with each story, **Actual Equivalent Story Points** is the actual story points a user story took at the end of the sprint as compared to the estimated in the beginning of sprint. This helps to improve the estimation techniques for future projects. **% Completed** indicates how much work is completed so far for each user story.

Story points are assigned at the start of each sprint for those user stories that are assigned to that sprint. The default is to use values of 1, 2, 3, 5, 8, 13, and 21. Anything bigger than 21 should be made into multiple stories. Each team should benchmark their story points and indicate what was used as the benchmark. For example, in an online bidding system, everyone in the group agrees that the *“Search for an item”* feature is 2 story points, so the points for all other stories are allocated in relation to that benchmark.

|  |  |  |
| --- | --- | --- |
| **ID** | **Feature name** | **Story points** |
| 5 | Register on the site | 3 |
| 4 | Place an item up for bid | 3 |
| 10 | Bid on an item | 3 |
| 17 | Auction engine | 8 |
| 13 | Search for an item | 2 |
| 16 | Purchase an item immediately | 2 |

**Acceptance Criteria** should define the boundaries of a user story and are used to confirm when a story is completed and working as intended. The **ID** column specifies the ID for the acceptance criteria with respect to the corresponding user story ID. The **Description** column specifies the acceptance criteria formally. The **Verification** column provides information about how the acceptance criteria will be tested.

Remember that not all user stories are necessarily functional. User stories can also specify the need for security (e.g. who can and cannot use a system), globalization (e.g. descriptions and other strings will not be hard coded so that the software can be translated into other languages more easily), portability (e.g. developing an iPhone application to be compatible with other smart phone platforms like Android or Windows Phone) accessibility (e.g. the program complying with ADA guidelines), availability (e.g. the system will be able to reboot in under 2 minutes in the event of failure.), etc.

**Tasks** consists of actual product development steps for each user story. The **ID** column specifies the task for a user story ID. The **Resource** column specifies the name of team member associated with a corresponding task ID >>

**Sample User Stories:** *This sample assumes SPRINT 2 is completed. For SPRINT 3, user stories are already planned along with* ***Acceptance Criteria*** *and* ***Verification Description****.* ***Tasks*** *will be added when SPRINT 3 is completed.*

**SPRINT 1**

**Estimated User Story Points:** 2

**Actual Completed User Story Points:** 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Added** | **Description** | **Status** | **Story Points** | **Actual Equivalent Story Points** | | **% Completed** |
| 100 | Onset | **As a** conference attendee,  **I want** to be able to register online,  **So that** I can register quickly and cut down on paperwork | **C** | **2** | **3** | | **100%** |
| **Acceptance Criteria** | | | **Verification** | | | | |
| **110** | A user cannot submit a form without completing all the mandatory fields | | **Create a test case to verify non-empty fields.** | | | | |
| **111** | Information from the form shall be stored in the registration database after form submission | | **Create test case to verify information is stored in the database.** | | | | |
| **112** | Payment shall be accepted via credit card | | **Create test case to verify credit card payment method from bank.** | | | | |
| **113** | An acknowledgment email shall be sent to the user after submitting the form. | | **Create test cases to verify sending of acknowledgement email after successful payment.** | | | | |
| **ID** | **Tasks** | | | | | **Resource** | |
| 1 | Create a registration page with all required fields (FName, LName, organization, Address details, email, credit card details) and register button at the bottom. | | | | | **Team member 1** | |
| 2 | Develop a backend functionality that checks required fields are non-empty when user clicks on register button. | | | | | **Team member 2** | |
| 3 | Built a functionality which verifies payment from bank based on credit card details at 1. | | | | | **Team member 2** | |
| 4 | Built a functionality which stores registration data in the database according to the specifications and sends acknowledgement email to the registered email else display payment failure message. | | | | | **Team member 3** | |

**SPRINT 2**

**Estimated User Story Points:** 8

**Actual Completed User Story Points:** 8

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Added** | **Description** | **Status** | **Story Points** | | **Actual Equivalent Story Points** | **% Completed** |
| 200 | Onset | **As a** customer,  **I want** to withdraw cash from an ATM  **So that** I don't have to wait in line at the bank. | **C** | **8** | | **8** | **100%** |
| **Acceptance Criteria** | | | **Verification** | | | | |
| **210** | ATM card shall be valid to be accepted | | **Create test cases to verify ATM card details with the banking system.** | | | | |
| **211** | ATM shall only try to dispense cash if it has enough to cover the amount requested | | **Create a test case to check cash in ATM dispenser.** | | | | |
| **212** | When customer requests cash, saystem shall ensure that amount withdrawn <= accountBalance | | **Usability test to determine if customer can enter required cash in the system.**  **Create test cases to check entry of valid amount.** | | | | |
| **213** | System shall ensure that account is debited, cash is dispensed and card is returned. | | **Create test cases to verify the following steps for a transaction: account data is modified, dispenser dispenses the cash, and card is returned back.** | | | | |
| **ID** | **Tasks** | | | | **Resource** | | |
| 1 | Develop a method which reads card details (card #, account #, validity) from the ATM card in the machine and verifies details from the banking system. If details are not verified, card rejection message shall be displayed and the card shall be returned to the customer. | | | | **Team member 1** | | |
| 2 | The developed function shall also include a verification technique which confirms that the ATM machine contains enough cash for the withdrawal. | | | | **Team member 2** | | |
| 3 | Create cash withdrawal page with submit button for customers where they can enter the required amount for withdrawal and press the submit button. | | | | **Team member 2** | | |
| 4 | Build a method which shall verify that the amount entered <= the amount available in the account else displays rejection without any transaction and returns card to the customer. | | | | **Team member 3** | | |
| 5 | Create a function which shall complete the transaction successfully in the following steps:   1. accountBal = accountBal – amountWithdrawn 2. Cash shall be dispensed by the ATM machine.   Card shall be returned to the customer | | | | **Team member 3** | | |

**SPRINT 3**

**Estimated User Story Points:** 55

**Actual Completed User Story Points:** N/A

**Main User Story:** “As the HR manager, I want to create a screening quiz so that I can understand whether I want to send possible recruits to the functional manager.”

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Added** | **Description** | **Status** | **Story Points** | **Actual Equivalent Story Points** | **% Completed** |
| 300 | Onset | **As a** manager,  **I want** to browse my existing quizzes  **So that** I can recall what I have in place and figure out if I can just reuse or update an existing quiz for the position I need now. | **C** | **8** |  |  |
| **Acceptance Criteria** | | | **Verification** | | | |
| **310** | Manager shall be able to search by quiz name. | | **Create test cases to verify search results by quiz name.** | | | |
| **311** | Manager shall be able to search by quiz topics included. | | **Create test cases to verify search results by quiz topics.** | | | |
| **312** | Manager shall be able to search by creation and last used date. | | **Create test cases to verify search results by creation and last used date.** | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Added** | **Description** | **Status** | **Story Points** | **Actual Equivalent Story Points** | **% Completed** |
| 400 | Onset | **As an** HR manager,  **I want** to match an open position’s required skills with quiz topics  **So that** I can create a quiz relevant for candidate screening. | **C** | **13** |  |  |
| **Acceptance Criteria** | | | **Verification** | | | |
| **410** | Manager shall be able to search the quiz topics by name. | | **Create test cases to verify search quiz topics by name.** | | | |
| **411** | Quiz topics shall be allowed to have alternate names, terms for searching | | **Create test cases to verify alternate quiz names and terms to search.** | | | |

* 1. Design diagrams/ UML diagrams/ Flow Charts/ E-R diagrams

<<Provide all the design diagrams that were created during the design phase of your project. Some examples are in sections below:>

* + 1. Use Case Diagrams

<< Provide any use-case diagrams that are being used as part of the project. Uniquely label each use case so that if necessary, it is easy to reference from other parts of the document. >>

**Example:** Restaurant system

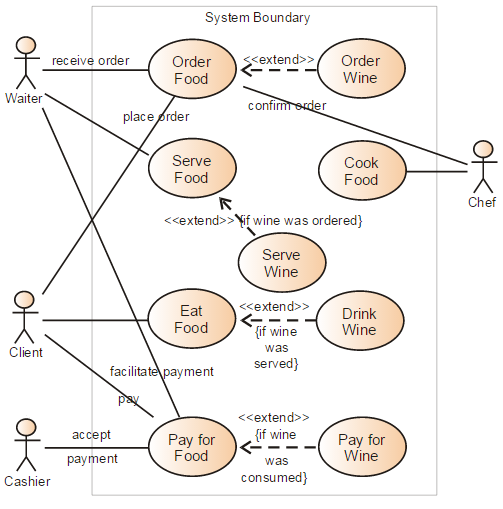


Figure 1: Sample use-case diagram

<<to insert caption for tables and figures, click on references on the top menu, under captions section 🡪 click insert caption 🡪 choose label as table

If you use a table, figure, or non-text item that is not your original design, you must cite the original source of the item. You may use an in-text citation in the text of the title or caption of the item, or you may include the citation as a footnote. Refer to the style manual of your discipline for more information about citations of non-text items.>>

* + 1. Class Diagram

<< Include a class diagram for all classes to be designed. Optionally include major data elements of those classes and important methods and functions that will be used by other classes.>>

**Example:** Online Photo Collection

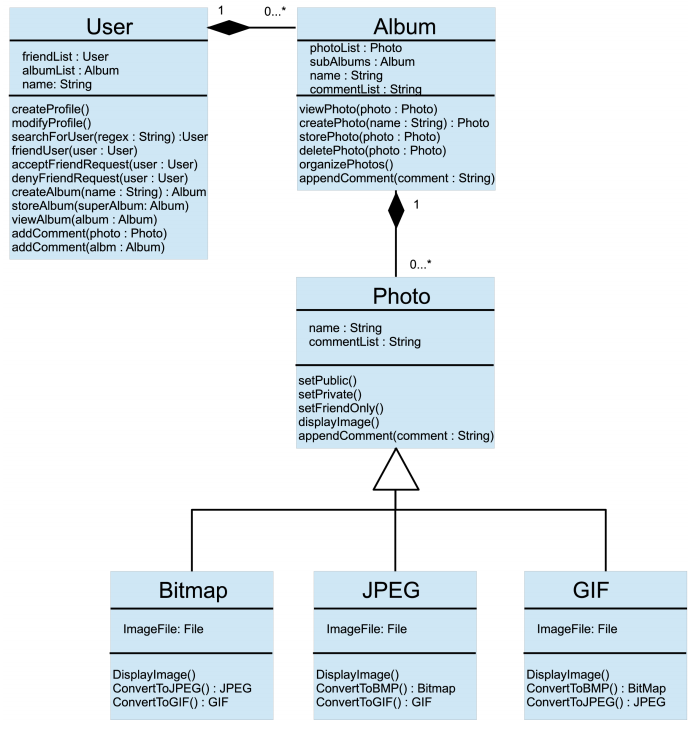


Figure 2: sample 2

* + 1. Activity Diagrams

<< Include activity diagrams for important workflows in the program. At least one diagram should be included for the main workflow in the program. Optionally include labels that indicate which component is responsible for that part of the activity. Activity diagrams for components which perform complex tasks should also be included. >>

**Example**: View friend’s photo album

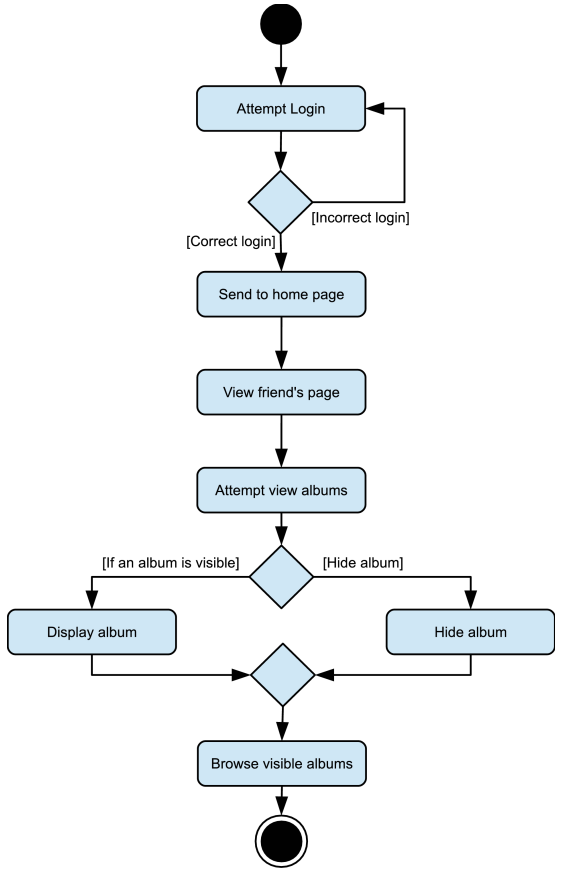


Figure 3: sample 3

* + 1. Sequence Diagram

<< Include sequence diagrams for important functionality of the program to indicate control flow. These diagrams should include classes found in the class diagram and use the methods for those classes to show the interaction between them. >>

Example: Create new album

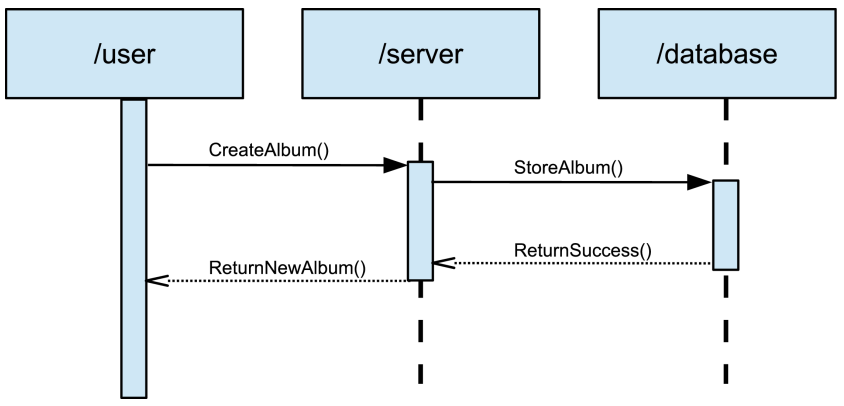


Figure 4: Sample 4

* + 1. Data Architecture

<< Include any information or diagrams that provide details about databases, xml configuration files, or other data structures that are a part of the system. If a very specific format is required, it may be worthwhile to provide a more robust description or a detailed design such as a database schema. >>

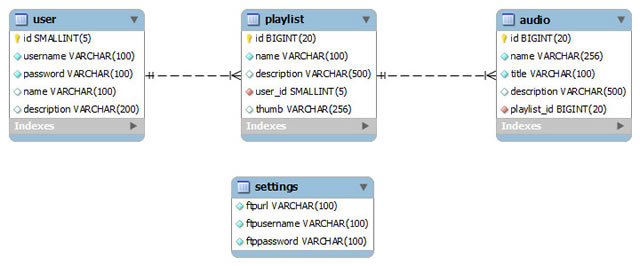


Figure 5: Sample 5